

IMPORTANTE:

- ❑ Crear un proyecto con su Nombre y Apellido.
 - ❑ Añadir comentarios a su código identificando con el número de ejercicio e inciso tanto a la/s función/es vinculadas a la resolución como a la porción de código del Main correspondiente.
- NO SE CORREGIRÁ CODIGO NO IDENTIFICADO.**

Una empresa que patrocina a distintos Youtubers con canales dedicados a juegos nos ha encomendado realizar un pequeño sistema.

Los datos para trabajar se encuentran almacenados en el archivo subido al campus, el cual hay que descargar y copiar dentro de la carpeta del proyecto del Parcial. Se trata de un archivo de estructuras "stYoutuber" que responde a la siguiente estructura de datos:

```
typedef struct {
    int id;
    char nombreCanal[40];
    char rubro[30];           //los rubros de juegos a los que se dedica cada canal son (y estan escritos
                             // tal cual: "fantasia", "creativos", "accion", "aventura", "retro", "construccion"
    int cantSuscriptores;     //cantidad de suscript total del canal
    int cantVistasSemestre;   //cantidad de vistas que recibieron sus videos en el último semestre
} stYoutuber;
```

Se nos pide desarrollar la siguiente funcionalidad:

1. Hacer una función que recorra el archivo adjuntado y cargue los datos en un ARREGLO de ARBOLES (ADA), clasificando los datos de la siguiente manera:

<p>En cada CELDA DEL ARREGLO se ubica c/u de los RUBROS, de acuerdo a la siguiente struct:</p> <pre>typedef struct{ char rubro[30]; }stRubro;</pre> <p>(CREAR LA STRUCT QUE CORRESPONDA PARA EL ARREGLO)</p>	<p>En cada celda del arreglo también se encuentra un ARBOL con todos los YOUTUBERS DEL RUBRO, de acuerdo a la siguiente struct:</p> <pre>typedef struct{ int id; char nombreCanal[40]; int cantSuscriptores; int cantVistasSemestre; } stYoutuber;</pre> <p>(CREAR LA STRUCT QUE CORRESPONDA PARA EL NODO DEL ARBOL)</p>
---	---

Deberán codificarse las funciones que sean necesarias para:

- Copiar los datos correspondientes a la estructura stRubro y a la estructura stYoutuber.
 - Buscar el rubro al cual corresponde el Youtuber a cargar en el ADA.
 - Agregar un nuevo rubro si éste no existe.
 - Agregar un nuevo Youtuber al rubro que corresponda (**insertar en el árbol en orden alfabético de acuerdo al campo nombreCanal**)
 - **Modularizar adecuadamente pensando en la responsabilidad de cada TDA**
- (30 puntos)

2. Realizar las funciones necesarias para **MOSTRAR** todos los elementos de la **estructura compuesta (ADA)**, permitiendo visualizar cada rubro con sus youtubers.
 - **Los Youtubers de cada rubro (el árbol) deberán ser mostrados EN ORDEN ALFABETICO (pensar qué modo de mostrado utilizar).**

- Para ahorrar tiempo, ya se nos ha dado la porción de código correspondiente al mostrado por pantalla de los datos de un RUBRO:

```
printf("\n ***** RUBRO ***** \n");
printf("\n .....%s..... ",p.rubro);
printf("\n ***** \n");
```

- y de un YOUTUBER :

```
printf("\n ----- YOUTUBER ----- \n");
printf("Id del Youtuber.....: %d ",p.id);
printf("\nNombre del Canal.....: %s ",p.nombreCanal);
printf("\nCantidad de Suscriptores en miles.....: %d ",p.cantSuscriptores);
printf("\nCantidad de vistas del semestre en miles.....: %d \n",p.cantVistasSemestre);
```

Modularizar adecuadamente pensando en la responsabilidad de cada TDA.

(8 puntos).

3. Realizar las funciones necesarias para **averiguar cuál fue el rubro con más cantidad de vistas por semestre del ADA**. Para ello, deberá modularizarse:

- una función que sume y retorne las cantidades de vistas por semestre de todos los youtubers de UN rubro.
- una función que analice TODOS los rubros, y encuentre y retorne el rubro que tuvo mayor cantidad de vistas semestrales.

(18 puntos).

4. Realizar las funciones necesarias para poder **calcular el promedio de vistas por semestre de todos los Youtubers del ADA que superen cierta cantidad de suscriptores** a elección del usuario (por ejemplo: el promedio de vistas de los youtubers de más de mil suscriptores correspondientes a cualquier rubro)

- **Modularizar adecuadamente pensando en la responsabilidad de cada TDA.**

(18 puntos)

5. Realizar las funciones necesarias para **recorrer el archivo adjuntado y cargar los datos en una LISTA DOBLEMENTE ENLAZADA de Youtubers** (debe utilizarse la estructura de datos del archivo completa, tal como está, no desglosada en Rubro y Youtuber como en los ejercicios anteriores).

- **Los datos deben ser agregados en la lista de la forma en que resulte más eficiente realizarlo** (al principio, en orden o al final?)

(18 puntos).

6. Main:

- Cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
- Muestre los resultados cada vez que sea necesario.
- A fin de identificar cada inciso, comente su código indicando a qué apartado corresponde.

(8 puntos)